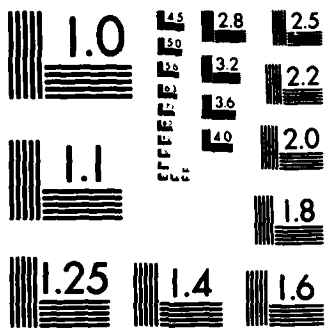
AD-A143 075 AN EXPERIMENTAL INTERNETWORK MULTIMEDIA MAIL SYSTEM(U) 1/1
UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY
INFORMATION SCIENCES INST A R KATZ JUN 84
UNCLASSIFIED ISI/RS-84-134 MDA903-81-C-0335 F/G 17/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

ISI Reprint Series
ISI/RS-84-134
June 1984

AD-A143 075

University of Southern California

Alan R. Katz

An Experimental Internetwork Multimedia Mail System

Reprinted from *Computer Message Services.* Proceedings of the IFIP 6.5 Working Conference, Nottingham, England, May 1984.

INFORMATION SCIENCES INSTITUTE ISI
213/822-1511
4676 Admiralty Way/Marina del Rey/California 90292-6695

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ISI/RS-84-134 ✓	2. GOVT ACCESSION NO. AD-A143 075	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Experimental Internetwork Multimedia Mail System		5. TYPE OF REPORT & PERIOD COVERED Research Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Alan R. Katz		8. CONTRACT OR GRANT NUMBER(s) MDA903 81 C 0335
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute 4676 Admiralty Way Marina del Rey. CA 90292-6695		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE June 1984
		13. NUMBER OF PAGES 15
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document is approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

....

18. SUPPLEMENTARY NOTES

This paper is reprinted from *Computer Message Services*, Proceedings of the IFIP 6.5 Working Conference. held in Nottingham, England, in May 1984.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

bitmap, bitmap display, digital voice, electronic mail, internet, internet message, multimedia mail, PERQ

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A description is given of an experimental multimedia mail system which operates within the DoD Internet Environment. The system runs on a PERQ personal computer and allows the user to compose, send, receive, read, and edit multimedia messages which contain text, image (bitmap), and voice data.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ISI Reprint Series
ISI/RS-84-134
June 1984

University of Southern California

Alan R. Katz

An Experimental Internetwork Multimedia Mail System

Reprinted from *Computer Message Services*, Proceedings of the IFIP 6.5 Working Conference, Nottingham, England, May 1984.

A-1

INFORMATION SCIENCES INSTITUTE
213/822-1511
4676 Admiralty Way/Marina del Rey/California 90292-6695

This research was supported by the Defense Advanced Research Projects Agency under Contract No. MDA903 81 C 0335. The views and conclusions contained in this .port are the author's and should not be interpreted as representing the official opinion or policy of DARPA, the U.S. Government, or any person or agency connected with them.

ISI Reprint Series

This report is one in a series of reprints of articles and papers written by ISI research staff and published in professional journals and conference proceedings. For a complete list of ISI reports, write to

Document Distribution
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
USA

An Experimental Internetwork Multimedia Mail System

Alan R. Katz

Information Sciences Institute
University of Southern California
Marina Del Rey, California
U.S.A.

Abstract:

A description is given of an of an experimental multimedia mail system which operates within the DoD Internet Environment. The system runs on a PERQ personal computer and allows the user to compose, send, receive, read, and edit multimedia messages which contain text, image (bitmap), and voice data.

I. Introduction

This paper describes the implementation and use of an experimental multimedia mail system, in particular the user interface program called MMM. Using MMM, it is possible for a user to create a multimedia message which may contain various types of text, image, and voice data and to then send the message to other hosts within the Department of Defense (DoD) Internet Environment (the Internet).

MMM is written in Pascal and runs on a PERQ personal computer equipped with a large bitmap display, a local hard disk, and a connection to a 3 Mbit Ethernet. A CHI-5 LPC speech box connected to the PERQ's RS-232 port provides voice input and output.

Hardcopy pictures may be scanned by a Rapicom Facsimile machine, then translated into bitmaps for inclusion in multimedia messages. These bitmaps may be edited, or others created using a bitmap sketching program (which is also a part of MMM).

Section II of this paper briefly describes the DoD Internet and the family of protocols used in this environment. The physical data connections between the PERQ running MMM and the various networks used are also discussed. Section III describes the specific protocol used. This protocol allows general types of structured data to be transfered within the Internet. Section IV describes the subset of this protocol implemented in MMM and gives a detailed account of how MMM works and how one would use it to compose, send, and receive messages. Section V discusses our experience with this system and the problems encountered. Finally, Section VI outlines our plans for the future.

II. The DoD Internet Environment

For the past several years, the DoD's Defense Advanced Research Projects Agency (DARPA) has sponsored research on various types of computer networks, including the ARPANET, digital packet radio networks, and digital packet satellite networks. A family of protocols was developed to allow all hosts in the interconnected set of these networks (refered to as the Internet) to communicate with one another. (An interconnected set of networks such as this as been termed a *Catenet* [1].) In particular, the Internet Protocol (IP) [2] and the Transmission Control Protocol (TCP) [3] were developed.

Hosts in the Internet range from very small personal computers on local area networks to large mainframes with many users on long distance networks such as the ARPANET. Data communication between these hosts is built upon IP datagrams. Data transfer via datagrams is unreliable, and higher level protocols (such as TCP) are necessary to ensure reliable communication. Networks within the Internet are connected to one another via gateways which route datagrams through the various networks, from gateway to gateway, until they arrive at their destination. (More information on the model of the Internet may be found in reference [4].)

In addition to IP and TCP, the User Datagram Protocol (UDP) [5] allows users to send IP datagrams. A simple file transfer protocol called the Trivial File Transfer Protocol (TFTP) [6] has been developed to allow the transmission of files using UDP.

At USC/Information Sciences Institute (ISI), we have three PERQ personal computers which are connected to a 3 Mbit Ethernet. We also have six TOPS-20 PDP-10 KL computers which are on the ARPANET. Multimedia messages created on the PERQs are transfered to the TOPS-20 system using TFTP through a PDP-11 gateway. These machines and networks are shown in figure 1.

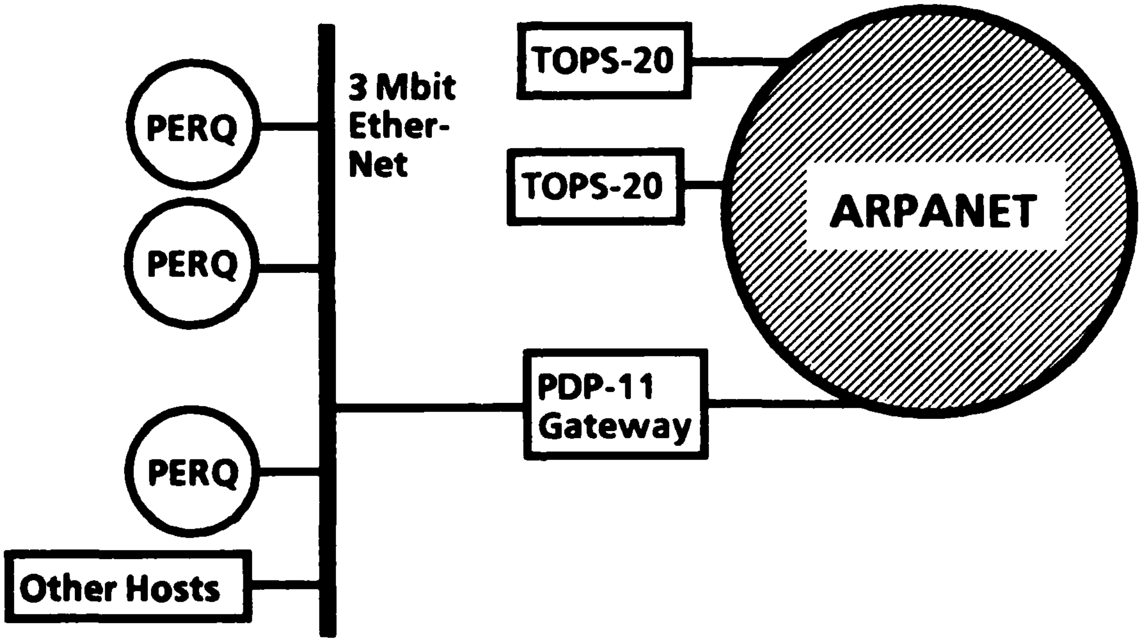


Figure 1: Connection of PERQs to ARPANET at ISI

III. The Internet Message System

An Internet Multimedia Mail Transfer Protocol [7,8] has been developed at ISI which defines a mechanism for the transfer of messages through the Internet. These messages may contain very general types of data including text, voice, image, facsimile, and graphics. The protocol is general enough to allow additional types of data to be added in the future.

The protocol is implemented in a process called a Message Processing Module (MPM) The MPMs are responsible for the routing, transmission, and delivery of messages. Multimedia messages are created by a user with a User Interface Program (UIP). Messages so created are then submitted to an MPM for delivery. At ISI, MMM is our UIP and runs on the PERQs. The MPMs, written in BLISS and some Macro10, run on PDP-10 TOPS-20 machines.

The Multimedia Mail Transfer Protocol assumes a reliable method of data transmission within the Internet and is therefore built upon TCP. The place of this protocol in the normal hierarchy of Internet protocols is depicted in figure 2.

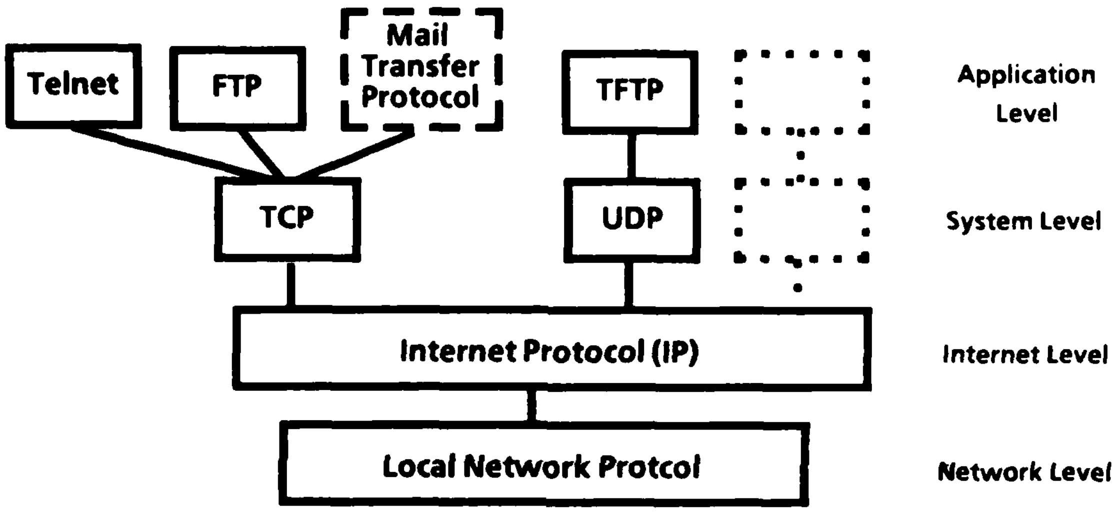


**Figure 2:
Relationships of Internet Protocols**

The basic unit transfered between MPMs, called a message, consists of three parts: a transaction identifier, which uniquely identifies the message, a command part, and the document. The document contains a header and a body.

The command part of the message contains information used by the MPM to route the message. Some of this information is supplied by the UIP. The document is not looked at by the MPM, but is displayed to the user by the UIP.

The header portion of the document corresponds to the date, to, from, etc., fields of a typical letter or inter-office memo. The body of the document contains the actual data of the message This data is a structure of basic data types (as defined in reference [7]). There are types for exactly representing integers, strings, booleans, etc. Two other elements are used for building data structures: the list and the property list. Lists are simple lists of elements (which may include other lists). Property lists are lists of pairs of elements, where the first element of each pair names the pair. The names of the name/value pairs in a property list are required to be unique.

References [9] and [10] describe the format of the document of a multimedia message. The body of the document may be a simple character string or a complex structure of lists and property lists.

It may be encrypted in part or in whole. Possible data structures allowed in the body include voice, paragraphs of text, and graphics.

The presentation of information in the message body can be compared to a seminar, where the speaker displays slides and other visual aids while providing a running commentary. The time coordination of such a presentation is captured in the body structure. There are three types of time ordered control possible within the document. They are: SEQUENTIAL, SIMULTANEOUS, and INDEPENDENT.

Sequential data items are presented one at a time, in the order listed. Simultaneous data is intended for synchronous viewing, and independent data can be presented in any time order. For example, one could have a text item displayed alone, then have graphics displayed while simultaneously listening to a voice description of the graphics.

Because of the very wide range of possible items within a document, we realized that in order to implement a UIP it would be necessary, at least at first, to limit this range. This is described in the next section.

IV. The UIP Implementation: MMM

Introduction

In the summer of 1982, a first attempt was made at limiting the possible data types allowed in a document body for the purpose of demonstrating the Multimedia Message System [11]. It was decided that, for the intial series of experiments, the only media allowed would be TEXT, VOICE, and IMAGE. The only time control allowed would be SEQUENTIAL.

TEXT data would be structured as a list of paragraphs only. (This was later extended to allow four types of TEXT data: Paragraph, Enumerate, Itemize, and Verbatim; roughly corresponding to commands in the text formatting system Scribe [12]). VOICE data would be LPC (Linear Predictive Coding, a way of digitizing voice [13,14]) only. The LPC data would be represented by a bitstream data type which represents a stream of speech data at 2400 bps. IMAGE data would be represented by raw bitmaps only (not compressed or encoded). It was suggested that these bitmaps not exceed 512 pels horizontally by 663 pels vertically (which is the same aspect ratio as 8 1/2 by 11).

It was expected that there would be at least two ways messages could be presented to the user. In the first way, there would be separate bitmap and text windows which would scroll separately. In the second way, both bitmap and text would be displayed in the same window in the order given and would scroll or page as necessary. In either case, voice data would not take up any space on the screen, and if a voice element appeared after an image element, it would be intended that the image be displayed while the voice was played.

The first approach was taken in the PERQ implementation, MMM, at ISI. The following subsections describe the program in detail.

Hardware

MMM is implemented on a PERQ Systems (previously Three Rivers Computer Corporation) PERQ personal computer. The PERQ has a 16-bit microprogrammed processor, a high resolution 1024 by 768 bit mapped raster display, a 12 megabyte Winchester hard disk, a megabyte of memory, and a tablet. It is also equipped with a standard RS-232 I/O interface.

The PERQ is optimized to be programmed in Pascal. Software is provided to make it easy to access the display (which is very fast) and for multiple window management.

Voice functions are provided by connecting the PERQ's RS-232 port to a CHI-5 Voicebox. The CHI-5 is a programmable array processor. It needs to be downloaded with a program before it

is able to perform voice I/O. The CHI-5 is connected to a speaker and a microphone. Later, we also used a Lincoln Labs voicebox, which does not need to be downloaded.

The PERQ is connected to a 3 Mbit Ethernet as described in Section II. Using TFTP, it can send multimedia messages to the MPM running on the TOPS-20 systems. The interconnection of the MMM hardware is shown in figure 3.

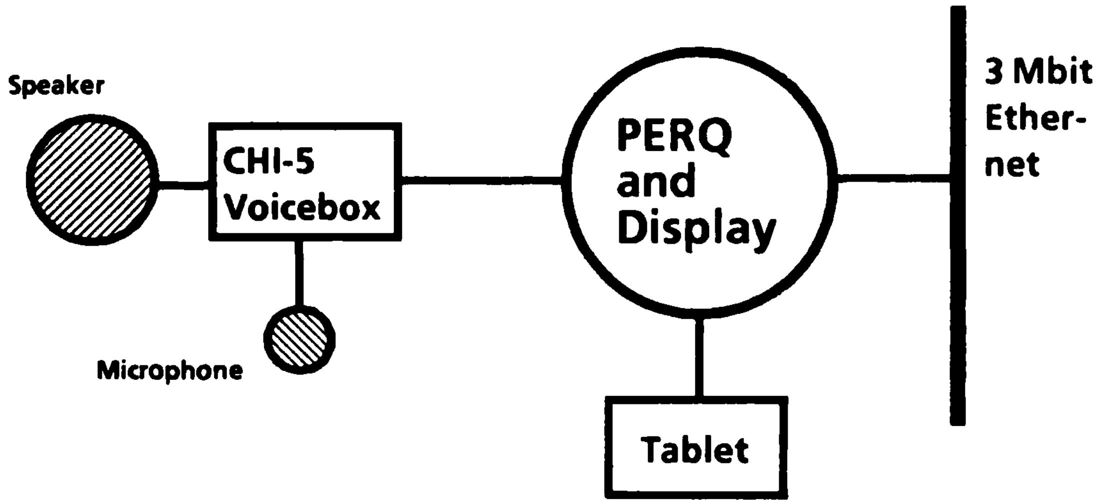


Figure 3:
Hardware Connections to the PERQ

Data Representation

In general, a message can be very large. One reason for this is that there can be a very large amount of data in a bitmap or voice data element. Thus, an entire bitmap may not fit into the availible PERQ memory and must therefore be stored in a file.

In MMM, the various parts of a message are stored in seperate files. The message structure, which contains pointers to the bitmap and voice data files, is itself stored in a file. This internal representation differs from the standard message representation, in which the entire message (including bitmaps and voice data) consists of one continuous stream of bits. MMM must therefore convert messages to and from the standard representation in order to send or receive messages using the MPM. This conversion process is transparent to the user, although reading in or sending out messages might take longer than the user would expect.

Because bitmaps and voice data are stored in separate files, programs other than MMM may access and operate on them without knowing anything about the Multimedia Mail Protocol. For example, it is possible to display and edit a stored bitmap using the bitmap editor.

Bitmaps

There are two ways to create a bitmap for inclusion in a multimedia message. The first is to scan an existing image with a facsimile machine, and the second is to create a bitmap with a bitmap

editor. The bitmap editor (which is a greatly modified version of a program supplied with the PERQ) can be run separately or under MMM.

At ISI, we use a Rapicom 450 facsimile machine. Programs exists both on the TOPS-20 systems and on the PERQ to store and send facsimile images from the Rapicom into a file and to convert that file into a standard bitmap [15,16] which can then be edited or viewed with the bitmap editor. The bitmap editor allows the user to view part or all of a bitmap, to draw an image using various brushes (analogous to using pens with differently shaped points), and to copy, move, or erase portions of the bitmap. It is also possible for a user to enter text in various fonts into the bitmap. If the bitmap is too large to fit into the display, the user can edit parts of it at a time.

A standard Rapicom 450 (8 1/2 by 11 inch) fascimile page, converted to a bitmap is 1726 by approximately 2200 bits (corresponding to 200 bits per inch). Since the display on the PERQ is 768 by 1024 bits, this page cannot be fully displayed unless it is compressed. The bitmap editor can perform this function.

Voice-related functions

If the CHI-5 Voicebox is used for voice I/O, the user must first download it with a special program. This takes a few minutes. Then, to enter voice, the user gives the appropriate command to MMM and simply speaks into the microphone. When finished speaking, the user types control-C to terminate input. Similarly, to listen to voice data, the user gives the command to MMM and the speech data is heard from the CHI-5 speaker.

Because of the time it takes for the PERQ disk to access a page, all voice data must be read into memory before being sent to the CHI-5. Thus, there is a delay of a few seconds before voice output and after voice input while the file is being read or written.

How MMM looks to the User

When the user firsts runs MMM, four windows are displayed: the bitmap window, the information window, the command window, and the text window (see Figure 4). Commands are entered by *bugging* with the tablet in a menu in the command window, with additional information being provided with the keyboard. MMM can be in three Modes: Top Level, Outline Mode, and Create Mode. Each of these modes has its own list of possible commands, which are displayed in the command window.

Headers (each containing a brief summary of a message) are displayed in the information window. Whenever a new message is received or created, a new header for that message is displayed.

It is possible to Play a message by using the Play command and pointing to the header of the message to be played. When a message is played, information such as the To and From fields of the message is displayed in the bitmap window. Descriptions of the various entities (text, bitmap, or voice) are displayed in the information window. After the user has read this and has hit a carriage return, the text entities in the message are displayed in the text window, bitmaps are displayed in the bitmap window, and voice entities are played on the CHI-5. If the bitmap is too large to fit into the window, the user has the option of specifying which portion to view, the default being the middle portion. When the text fills up the text window, it is scrolled.

If the user wants to look at particular entities in the message out of order, he can go into Outline Mode, which allows the user to view particular entities and to edit or store them.

Messages may be created via Create Mode. In this mode, the user may enter the names and addresses of receipients, enter a subject field, and enter bitmaps, voice, or text data. Bitmaps may come from a file (which was created earlier with the bitmap editor or from the fascimile machine) or may be created on the spot. The same is true for voice or text data. When the user

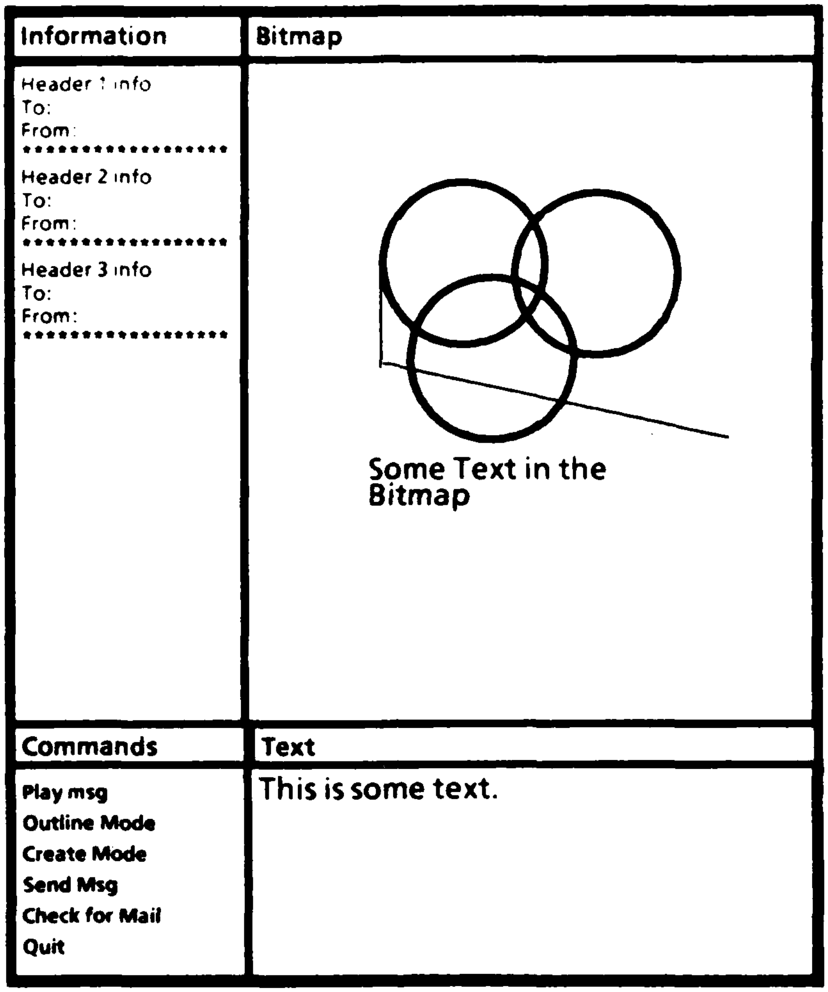


Figure 4:
The MMM Display

is satisfied with the composition of the multimedia message, he may leave Create Mode and the message is then created and stored. A header for that message will appear in the information window when the user is in Top Level Mode.

Sending and Receiving Messages: Communication with the MPM

In order to read in new messages which may be waiting, the user must explicitly tell MMM to read them in using the Check for New Mail command. The PERQ cannot continually check for new messages since the operating system does not provide for parallel processes to be run.

As was mentioned previously, the MPM at ISI runs on a TOPS-20 system on the ARPANET. Each MMM user must have a directory on that system into which the MPM will place new mail and from which the MPM will read messages to be sent. When the user uses the Check for New Mail command, MMM attempts to TFTP (see section II) a file containing that user's incoming mail. If the TFTP is successful, MMM converts the message into it's internal representation and then

displays a header for that message in the information window. If the TFTP is unsuccessful, MMM notifies the user that no new mail exists. If a new message is read in, the user may outline or read the message as described in the previous subsection.

Similarly, after creating a message, the user must use the Send command to actually send the message. Then, MMM converts from the MMM internal format for the message and TFTPs the message to the TOPS-20 host, where the MPM will pick it up and send it on its way. MMM also sends a UDP datagram to a server on the TOPS-20 system which tells the MPM to look for new mail in the specified directory for that user (the MPM also checks all directories for new mail by itself at specific intervals).

V. Experience and Results

In October 1982, we demonstrated the MMM/MPM system by composing multimedia messages containing various types of text, bitmaps, and voice data; sending them to our MPM which then delivered them to their destinations; and receiving and displaying the multimedia replies. We were able to send messages to Bolt Beranek and Newman, Inc. (BBN) and the Massachusetts Institute of Technology (MIT) in Cambridge, Massachusetts, and to receive replies from BBN (MIT was able to view our messages but was unable, at the time, to create messages of its own). In one particular message, we sent a bitmap and some voice data to BBN and they replied with a bitmap depicting how our bitmap looked as displayed on their system.

We have recently been running MPMs on two TOPS-20 hosts and have been sending multimedia mail routinely using both of them to and from the three PERQs at ISI

A number of problems were encountered with the PERQ, in particular with the RS-232 interface. When one alternates between sending and receiving data through the RS-232 port (as one must do to interface with the CHI-5 Voicebox), the port will sometimes stop sending data, and will send out many ASCII nulls. This destroys the downloaded program in the CHI-5 and the user must stop and redownload. Even when this does not happen, other problems with the RS-232 port cause voice I/O to be somewhat tricky and unreliable.

Also, while it is possible to perform both speech input and output using a special program separate from MMM, when the program was incorporated into MMM (which is a rather large program), the processor became too slow to keep up with incoming voice. Thus one may listen to voice in MMM, but one must use a separate program to create a voice data file.

On the other hand, the bitmap display and editing parts of MMM work extremely well. The display is very fast, considering its high resolution. Other than the problems encountered with voice, MMM seems to work well and is regularly used in the Internet Project at ISI

VI. Future Plans

Because of the problems encountered with the PERQ and because ISI plans to utilize personal workstations (probably Xerox 8010 processors) to a much greater extent in the future, we have decided to implement the next version of a UIP on a Xerox 8010. The Xerox system allows multiple processes, and presumably we will not encounter the RS-232 problems that have plagued us on the PERQ.

The MPM continues to run on our TOPS-20 systems and at several other sites on the ARPANET. This makes it possible to exchange multimedia messages with other organizations that implement UIPs.

Recently, a system has been written to use a Packet Voice Terminal (PVT), connected to the Internet, as a voice server. When this system is incorporated into MMM, the PERQ will be able to send packets containing voice data to the PVT. The PVT will then call the user on his telephone and play the voice through the receiver. A similar chain of events will allow the user to enter voice data. This system will eliminate the need for each PERQ to have its own voicebox and will improve performance, as we will be able to bypass the RS-232 port.

VI. Acknowledgements

This research was done as a part of the ISI Internetwork Concepts Research Project and grateful acknowledgement is made to Jon Postel for his guidance and suggestions concerning this work. The MPM was programmed by Greg Finn.

References

[1] Pouzin, L., "A Proposal for Interconnecting Packet Switching Networks," Proceedings of EUROCOMP, Bronel University, May 1974.

[2] Postel, J.,"Internet Protocol", RFC 791, USC Information Sciences Intitute, September 1981.

[3] Postel, J., "Transmission Control Protocol", RFC 793, USC Information Sciences Intitute, September 1981.

[4] Cerf, V. "The Catenet Model for Internetworking", IEN 48, Defense Advanced Research Projects Agency, July 1978.

[5] Postel, J., "User Datagram Protocol", RFC 768, USC Information Sciences Intitute, August 1980.

[6] Sollins, K., "The TFTP Protocol", RFC 783, MIT Laboratory for Computer Science, June 1981.

[7] Postel, J., "Internet Multimedia Mail Transfer Protocol", RFC 759-revised, USC Information Sciences Intitute, March 1982.

[8] Postel, J., "An Internetwork Message Structure", Sixth Data Communication Symposium, November 1979.

[9] Postel, J., "Internet Multimedia Mail Document Format", RFC 767-revised, USC Information Sciences Intitute, March 1982.

[10] Finn, G. and Postel, J., "Data Structures and Presentation Control for Multimedia Computer Mail", Proceedings of EASTCON, November 1981.

[11] Katz, A., "Proposal for the Document Format for a Multimedia Message Experiment", ISI Internal Memo, July 1982.

[12] Reid, B. and Walker, J., "Scribe - Introductory User's Manual", Unilogic Ltd., May 1980.

[13] Hofstetter, E., Tierney, J., and Wheeler, O. "Microprocessor Realization of a Linear Predictive Vocoder", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-25, Number 5, October 1977.

[14] Malpass, M. and Feldma, J. "A User's Guide for the Lincoln LPC Speech Processing Peripheral", MIT Lincoln Laboratory Project Report PSST-2, April 1983.

[15] Katz, A., "Decoding Facsimile Data from the Rapicom 450", RFC 798, USC Information Sciences Intitute, September 1981.

[16] Katz, A., "Format for Bitmap Files", RFC 797, USC Information Sciences Intitute, September 1981.

NOTE: *In the above references, the term RFC refers to papers in the ARPA "Request for Comments" series and IEN refers to ARPA "Internet Experiment Notes." These notes may be obtained from the Network Information Center, SRI International, Menlo Park, California, or from the author.*

END

FILMED

8

DTIC